# An Improved Approximation Algorithm for the Traveling Salesman Problem with Relaxed Triangle Inequality[*]

Tobias Mömke

Saarland University, Saarbrücken, Germany
moemke@cs.uni-saarland.de

**Abstract**

Given a complete edge-weighted graph $G$, we present a polynomial time algorithm to compute a degree-four-bounded spanning Eulerian subgraph of $2G$ that has at most 1.5 times the weight of an optimal TSP solution of $G$. Based on this algorithm and a novel use of orientations in graphs, we obtain a $(3\beta/4 + 3\beta^2/4)$-approximation algorithm for TSP with $\beta$-relaxed triangle inequality ($\beta$-TSP), where $\beta \geq 1$. A graph $G$ is an instance of $\beta$-TSP, if it is a complete graph with edge weights $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ that are restricted as follows. For each triple of vertices $u, v, w \in V(G)$, $c(\{u, v\}) \leq \beta(c(\{u, w\}) + c(\{w, v\}))$.

## 1 Introduction

In the traveling salesman problem we are given a complete edge-weighted graph and we have to find a minimum-weight Hamiltonian tour, i. e., a tour that visits each vertex exactly once. This classical problem has been studied extensively and in many variations. Most of the variations concern restrictions of the weight function. One of the most natural restrictions is to assume that the weight function is a metric. Intuitively this means that we are allowed to take the shortest path to the next vertex to visit, even if this means to visit some of the vertices more than once.

Despite intensive research for more than 30 years, Christofides' algorithm is still the best known approximation algorithm for the metric traveling salesman problem and its approximation ratio is 1.5 [1]. For similar settings, however, a recent fast development has started. For graphic metrics—metrics obtained by taking the lengths of the shortest paths in an unweighted graph as weights—a

[*] This work was partially supported by ERC advanced investigator grant 226203 and Deutsche Forschungsgemeinschaft grant BL511/10-1

sequence of improvements was published within a short time frame, with the currently best approximation algorithm of Sebő and Vygen which achieves an approximation ratio of 1.4 [2, 3, 4, 5].

Instead of restricting the metric, in this paper we consider a relaxation. We use a relaxation parameter $\beta \geq 1$ and require that the input instance is a complete graph $G$ where the non-negative weight function $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfies the relaxed triangle inequality $c(\{u, w\}) \leq \beta(c(\{u, v\}) + c(\{v, w\}))$ for any three vertices $u, v, w \in V(G)$.

For approximation algorithms, the relaxed triangle inequality was introduced by Bandelt, Crama, and Spieksma [6]. This type of parameterization also provides a suitable type of relaxation in the context of stability of approximation [7] and our result fits into this framework.

The $\beta$-relaxed version of metric TSP ($\beta$-TSP) was first considered by Andreae and Bandelt [8] who presented a $1.5\beta^2 + 0.5\beta$ approximation algorithm. Subsequently Andreae improved the result to $\beta + \beta^2$ [9]. The next development was due to Böckenhauer et al. [7]. They obtained a $1.5\beta^2$-approximation algorithm, which is better than the previous algorithms for $1 < \beta < 2$. Bender and Chekuri [10] independently obtained a $4\beta$ approximation algorithm, which is better than the algorithm of Andreae for $\beta > 3$.

### 1.1 Results and Overview of Techniques

We provide an improved approximation algorithm for $\beta$-TSP.

**Theorem 1.1.** *There is a polynomial time $(3\beta/4 + 3\beta^2/4)$-approximation algorithm for $\beta$-TSP.*

The approximation ratio of our algorithm outperforms the ratios provided by Andreae [9] and by Böckenhauer et al. [7] for all values of $\beta$. The $4\beta$ approximation algorithm of Bender and Chekuri [10] still is better than our algorithm for $\beta > 13/3 \approx 4.33$. To obtain our result we first use the matroid version by Király et al. [11] of a bounded degree spanning tree result of Singh and Lau [12], combined with special $b$-matchings that respect parities of vertex degrees. We obtain a degree-4-bounded spanning Eulerian subgraph in $2G$ for any complete graph $G$ with edge weights $c\colon E(G) \to \mathbb{Q}$ such that the weight of the computed graph is at most 1.5 times the weight of an optimal TSP solution. Finally, we introduce an orientation technique that provides a cactus graph with useful properties such that the weight of shortcuts within the graph is restricted when constructing the TSP solution. One key insight is that we obtain two disjoint sets of edges such that we have to consider the factor $\beta^2$ only for the smaller of the two sets.

## 2 Preliminaries

All graphs in this paper are allowed to have multiple edges. For convenience of notation, however, we do not distinguish between multi-sets and sets. We handle multiple edges of a graph as separate edges. This way there may be cycles of length two. (We define a cycle to be a simple cycle, i. e., vertices may not be visited twice.)

Given a graph $G$, $V(G)$ and $E(G)$ are its set of vertices and its set of edges. For a set of edges $F \subseteq E(G)$, we write $c(F)$ as shorthand for $\sum_{e \in F} c(e)$. Similarly, for a graph $G$ we write $c(G)$ as shorthand for $c(E(G))$. A *block* of a graph $G$ is a maximal two-vertex-connected subgraph.

A $b$-matching of a graph $G$ is a subgraph $G'$ of $G$ with possible additional multiplicities of edges where each vertex has a degree of at most $b$. We identify a $b$-matching with its characteristic vector $\mathbf{x}$ of edges, that is, for each edge $e \in E(G)$, $\mathbf{x}$ has an entry $x_e \in \mathbb{N}_0$ and $x_e \geq 1$ if and only if $e \in E(G')$.

Let $G = (V, E)$ be an undirected graph and let $M$ be the $V \times E$ incidence matrix of $G$. Let $\mathbf{l} \leq \mathbf{m}$ and $\mathbf{a} \leq \mathbf{b}$ be integer vectors in $\mathbb{N}_0^E$ resp. $\mathbb{N}_0^V$ and let $S^{\text{odd}}$ and $S^{\text{even}}$ be disjoint subsets of $V$.

Consider the following constraints that impose restrictions on the characteristic vector of the $b$-matching $\mathbf{x} \in \mathbb{N}_0^E$.

$$\begin{array}{llllll} \text{(i)} & \mathbf{l} \leq \mathbf{x} \leq \mathbf{m} & \quad \text{(iii)} & (M\mathbf{x})_v \text{ is odd} & \text{if } v \in S^{\text{odd}} \\ \text{(ii)} & \mathbf{a} \leq M\mathbf{x} \leq \mathbf{b} & \quad \text{(iv)} & (M\mathbf{x})_v \text{ is even} & \text{if } v \in S^{\text{even}} \end{array} \tag{1}$$

These constraints specify (i) bounds on the multiplicity of edges in $G$, (ii) bounds on the degrees of the vertices in $G$, and (iii,iv) the parities of degrees for specific vertices (since $M\mathbf{x}$ is the vector of vertex degrees).

We slightly abuse notation and, whenever the meaning is clear from the context, we associate an integer $\mathbf{i}$ with the corresponding vector $(i, i, \ldots, i)$. Note that for an integer $b'$, (1) specifies a $b'$-matching if we set $\mathbf{l} = \mathbf{0}$, $\mathbf{m} = \boldsymbol{\infty}$, $\mathbf{a} = \mathbf{0}$, $\mathbf{b} = \mathbf{b}'$, and $S^{\text{odd}} = S^{\text{even}} = \emptyset$. For our results we need specific $b$-matchings that we will specify by giving values to the parameters of (1).

The following theorem is Theorem 36.5 in Schrijver's book [13]. (Here, we use a simplified setting. In the original theorem a more general class of graphs can be used.)

**Theorem 2.1** (Edmonds, Johnson [14]). *For any $\mathbf{c} \in \mathbb{Q}^E$, an integer vector $\mathbf{x}$ maximizing $\mathbf{c}^T\mathbf{x}$ over* (1) *can be found in strongly polynomial time (if it exists).*

Given a graph $G$, a 1-tree is a subgraph of $G$ composed of a spanning tree on $V(G) \setminus \{v_1\}$ for some $v_1 \in V(G)$ and two edges incident to $v_1$. If $G$ is a complete graph, then for each choice of $v_1$ there is a matroid $M$ such that the 1-trees of $G$ are the bases of $M$ [15]. We are interested in 1-trees where additionally the vertices have degree restrictions.

For a vector $\mathbf{b}$ of vertex degrees, a 1-tree $T$ of $G$ is degree $\mathbf{b}$ bounded if the degree of each vertex $v \in V(T)$ is at most $\mathbf{b}_v$. The following theorem follows directly from Király et al. [11] who showed a more general result for matroids.[1] (We run their algorithm for each choice of $v_1$.)

[1] Note that by guessing two consecutive edges and applying straightforward graph modifications, we could also use the predecessor result on bounded degree spanning trees [12].

**Theorem 2.2.** *Given a complete graph $G$ with edge weights $c\colon E(G) \to \mathbb{Q}$ and a vector $\mathbf{b}$ of upper bounds on the vertex degrees, there is polynomial time algorithm that computes a 1-tree $T$ in $G$ such that (a) $T$ is degree $(\mathbf{b}+\mathbf{1})$ bounded and (b) $c(T) \leq c(T')$ for all degree $\mathbf{b}$ bounded 1-trees $T'$ in $G$.*

For our algorithm we need cactus graphs. Here we use the following definition of cacti which strictly speaking specifies the subclass of 2-edge-connected cacti.

**Definition 2.3** (Cactus)**.** A graph $G$ is a *cactus* if it is 2-edge connected and all blocks are cycles, i. e., no two cycles share an edge.

Given a graph $G$, $2G$ is the graph where each edge of $G$ is doubled. A graph $G'$ is a *spanning Eulerian subgraph of* $2G$ if $V(G) = V(G')$, $G'$ is a connected subgraph of $2G$, and each vertex of $G'$ has even degree.

# 3 The Algorithm $\mathrm{Alg}_\beta$

Before we proceed to the main result, we show an intermediate observation. With the preparation in the preliminaries, the following theorem is not hard. It is, however, of independent interest given the importance of Eulerian graphs. Note that the weight function is neither required to be metric nor to be non-negative.

**Theorem 3.1.** *Let $G$ be a complete graph with edge weights $c\colon E(G) \to \mathbb{Q}$. There is a polynomial time algorithm that computes a degree-4-bounded spanning Eulerian subgraph of $2G$ such that its total weight is at most 1.5 times the weight of an optimal TSP solution in $G$.*

*Proof.* We combine Theorem 2.1 with Theorem 2.2. To obtain a degree-4-bounded Eulerian subgraph we compute a degree-3-bounded 1-tree $\tau$ and add a $b$-matching with the following parameters of (1). We set $\mathbf{l} = \mathbf{0}$, $\mathbf{m} = \mathbf{1}$, $\mathbf{a} = \mathbf{0}$, and $\mathbf{b} = \mathbf{2}$. Then we set $S^{\text{odd}}$ to be the set of all vertices of odd degree in $\tau$, and $S^{\text{even}}$ the set of all remaining vertices.

This way, all odd degree vertices in $\tau$ obtain a degree increased by exactly one and all remaining vertices (of degree two) either stay unchanged or become degree-4-vertices. By Theorem 2.1 we obtain an optimal $b$-matching $M$ with the specified parameters in strongly polynomial time.

Due to Theorem 2.2 we can, in polynomial time, obtain a degree-3-bounded 1-tree of at most the weight of an optimal TSP solution, because any TSP solution is a degree-2-bounded 1-tree. For the remaining argument we use that, similar to the matchings in Christofides' algorithm, any TSP solution is composed of two edge-disjoint $b$-matchings obeying the specified restrictions. A minimum weight solution thus has at most half the weight of an optimal TSP solution. □

In order to state the algorithm, we need a few more definitions. We use bi-directed arcs, where we mark the tails with + and the heads with −. Thus for an arc with two heads $u$ and $v$, we write $\{u^-, v^-\}$. For an arc with tail $u$ and head $v$, we write $(u, v)$ instead of $\{u^+, v^-\}$. In the context of this paper, there are no arcs $\{u^+, v^+\}$. Sometimes, we are interested in the vertices of arcs, independent of directions. In these cases, we refer to arcs as edges.

Let $A$ be the arc set and $V$ the vertex set of a bi-directed graph $D$. Let $v \in V$ be a vertex of $D$. The in-degree of $v$ is $\deg^-(v) = |\{a \in A : \exists u \in V, a = (u, v) \text{ or } a = \{u^-, v^-\}\}|$. Similarly, $\deg^+(v) = |\{a \in A : \exists u \in V, a = (v, u)\}|$.

We design an algorithm $\textsc{Alg}_\beta$ that relies on graph transformations specified by the following rule.

**Rule 1.** Let $D$ be a bi-directed graph and let $v$ be a vertex given as input such that $\deg^+(v) = 2$. Let $w \neq w'$ be the vertices such that the arcs $(v, w), (v, w')$ are in $D$. We remove both $(v, w)$ and $(v, w')$ from $D$ and add the arc $\{w^-, w'^-\}$.

We also need the reverse operation of Rule 1. If we obtained $\{w^-, w'^-\}$ due to an operation of Rule 1 to $v$, to reverse Rule 1 at $\{w^-, w'^-\}$ means that we remove $\{w^-, w'^-\}$ and add the two arcs $(v, w), (v, w')$.

Let $G'$ be a bi-directed graph. Suppose we are given a set of vertices $S \subset V(G')$ and vertices $w, w' \in S$, $v \in V(G') \setminus S$ with the following properties: (i) there are no arcs of $G'$ between $S$ and $V(G') \setminus S$; (ii) each connected component of $G'$ is a cactus; (iii) $w$ and $w'$ are connected by an arc $\{w^-, w'^-\}$ in $G'$, obtained due to an application of Rule 1 to $v$. Then we say that $\{w^-, w'^-\}$ is an *entry site* of the subgraph of $G'$ induced by $S$. Let $C$ be the block of $G'$ that contains $v$. Then after reversing the application of Rule 1 at $\{w^-, w'^-\}$, $v$ is an *exit point of* $C$ (but not of any other block).

Fig. 1(c),(d) shows an example of an entry site and an exit point. The reason to consider entry sites and exit points is to create a cactus such that its blocks form a specific hierarchy that allows us to exclude long (and therefore expensive) shortcuts when constructing a Hamiltonian tour from the cactus.

## 3.1 Proof of Theorem 1.1

We show that $\textsc{Alg}_\beta$ is a polynomial time $(3\beta/4 + 3\beta^2/4)$-approximation algorithm for $\beta$-TSP.

Let $G$ be an instance of $\beta$-TSP, where $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ is the cost function of $G$. We have to verify that $\textsc{Alg}_\beta$ indeed computes a Hamiltonian tour (see Fig. 1). Observe that due to the relaxed triangle inequality, $G$ is a complete graph; therefore we can apply Theorem 3.1 to obtain a degree-4-bounded spanning Eulerian subgraph $H$ of $2G$.

Since $H$ is Eulerian, we can efficiently find the required Eulerian orientation in $H$ by forming cycles of length two from parallel edges and iteratively adding further cycles. (The requirement to orient parallel edges in opposite directions prevents loops in the subsequent step and therefore simplifies the analysis.) We use the following property of $H$, observed by Euler.

**Claim 3.2.** *Let $S \subseteq V(H)$. Then $|\delta^+(S)| = |\delta^-(S)|$, where $\delta^+(S)$ and $\delta^-(S)$ are the sets of arcs from $S$ to $V(H) \setminus S$ resp. from $V(H) \setminus S$ to $S$.*

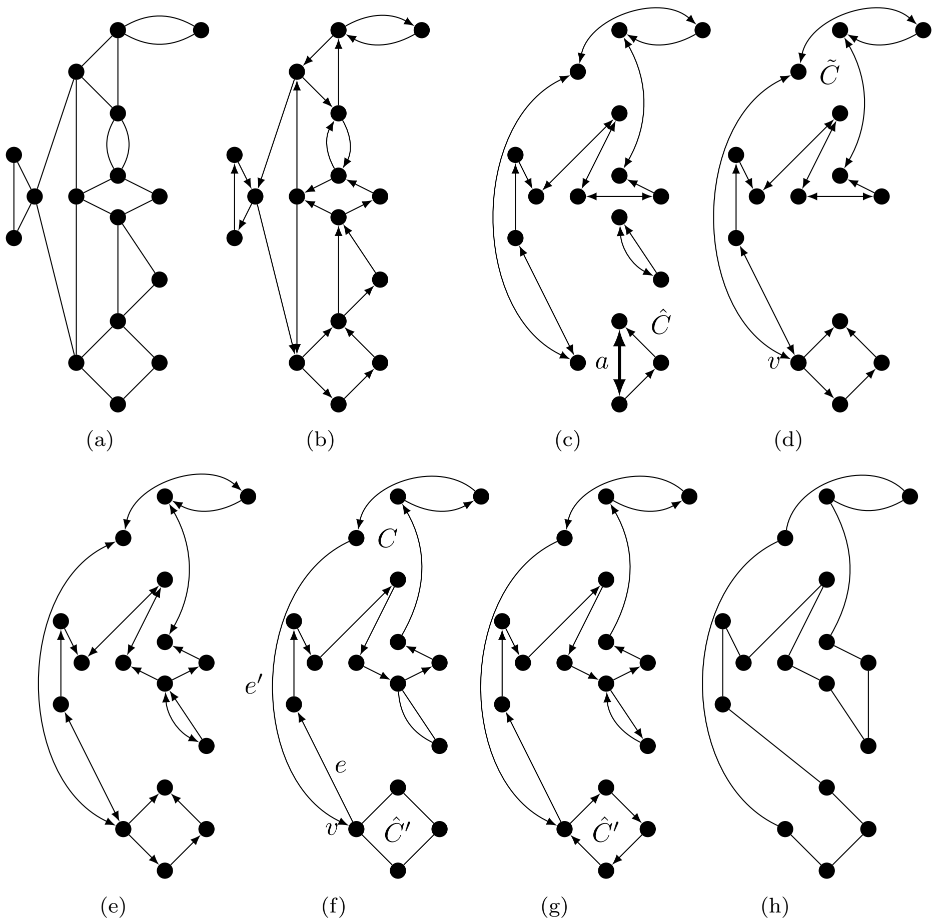
```
Algorithm: Alg$_\beta$
Input  : An instance $G$ of $\beta$-TSP.
Output: A Hamiltonian tour in $G$.
Apply Theorem 3.1 to obtain a degree-4-bounded spanning Eulerian subgraph $H$ of $2G$;
Find an Eulerian orientation in $H$ where parallel edges are oriented in opposite directions;
Apply Rule 1 to all vertices of degree 4 in $H$. Let $H'$ be the resulting graph ;     // The graph $H'$ is a collection of disjoint cycles.
Choose a cycle $\hat{C}$ in $H'$ and set $K = \hat{C}$;          // $K$ is a cactus.
while $V(K) \neq V(G)$ do
    Determine an entry site $a$ of $K$;
    Reverse the application of Rule 1 at $a$;
    Update $K$ to include the additional cycle as a new block;
end
Let $\hat{C}'$ be the block in $K$ that contains $V(\hat{C})$;
Let $K'$ be $K$ with all orientations of edges removed;
for each block $C$ of $K'$ except $\hat{C}'$ do          // Reorient the cactus
    Let $v$ be the exit point of $C$ (w. r. t. $K$);
    Let $e, e'$ be the two edges incident to $v$ in $C$ such that $c(e) \leq c(e')$;
    Orient the edges along $C$ in $K'$, starting with $e$ from $v$;
end
Orient the edges along $\hat{C}'$ in $K'$ (in arbitrary direction);
Apply Rule 1 to all vertices of degree 4 in $K'$;
Return the resulting tour $K''$ (without orientations).
```

In particular, Claim 3.2 implies that for each degree-4 vertex $v$ of $H$, $\deg^-(v) = \deg^+(v) = 2$. Note that applying Rule 1 to $v$ transforms $v$ into a degree-2 vertex and that the in- and out-degrees of all other vertices stay unchanged. Therefore we can apply Rule 1 to every degree-4 vertex of $H$ and afterwards each vertex has a degree of exactly 2, i. e., $H'$ is a collection of cycles.

Next we show that $K$ is always a cactus, and that in each run of the while loop we can find an entry site $a$. Initially $K$ is a cycle and therefore a cactus. Reversing Rule 1 at an entry site does the following. The cycle $C$ of $K$ that contains the entry site $a$ obtains an additional vertex $v$. The cycle in $H'$ containing $v$ stays unchanged, but now it shares $v$ with $K$. Both increasing the length of $C$ and attaching a cycle to $K$ that only shares a single vertex with $K$ preserve the properties of cactus graphs.

**Claim 3.3.** *Before each run of the while loop, $K$ has an entry site.*

*Proof.* Since $H$ is a connected graph, due to Claim 3.2 there are arcs from $V(H') \setminus V(K)$ to $V(K)$ in $H$. Let $\hat{a} = (v, w)$ be one of these arcs. Since we only removed arcs from $H$ by applying Rule 1, the degree of $v$ in $H$ is 4 and



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: An example of intermediate graphs in $\mathrm{ALG}_{\beta}$. (a) The graph $H$. (b) The Eulerian orientation in $H$. (c) The graph $H'$ before the while loop with entry site $a$ of $\hat{C}$. (d) The graph $K$ after one iteration of the while loop. The vertex $v$ is the exit point of $\tilde{C}$. (e) The graph $K$ after the while loop. (f) The graph $K'$ after one iteration of the for loop where $v$ is the exit point of $C$. (g) The graph $K'$ after the for loop. (h) The graph $K''$.

there is a second arc $\hat{a}' = (v, w')$ in $H$ that is not in $H'$. We conclude that there is an arc $\{w^-, w'^-\}$ in $K$. Since there are no arcs in $E(H')$ between $V(K)$ and $V(H) \setminus V(K)$ and $K$ is a cactus, we conclude that $\{w^-, w'^-\}$ is an entry site. □

By Claim 3.3, the while loop finds an entry site in each run and terminates with $V(K) = V(G)$. We now show that in the for loop, the vertex $v$ is well defined.

**Claim 3.4.** *After the while loop, each cycle of $K$ except $\hat{C}'$ has exactly one exit point.*

*Proof.* The cactus $K$ is formed by reversing applications of Rule 1 in order to add cycles. For each cycle $C$ that is added to $K$ in the while loop, there are vertices $w, w' \in V(K)$ and $v \in V(C)$ such that reversing the application of Rule 1 at $\{w^-, w'^-\}$ introduces the arcs $(v, w)$ and $(v, w')$. This means that $v$ is a cut vertex of the transformed $K$ and thus an exit point of $C$. Each cycle $C'$ of $H'$ except $\hat{C}$ is added to $K$ exactly once during the while loop and there is exactly one cycle $C''$ of $K$ with $V(C') \subseteq V(C'')$. We conclude that $C''$ has exactly one exit point. □

Therefore the for loop terminates. We now analyze the last step of $\text{A}\textsc{lg}_\beta$.

**Claim 3.5.** *$K''$ is a Hamiltonian tour.*

*Proof.* Let $j$ be the number of degree-4 vertices in $K'$ after the for loop. We observe that the order of applications of Rule 1 does not change the resulting graph. Therefore we may assign indices $1, 2, \ldots, j$ to the degree-4 vertices of $K'$ such that Rule 1 is applied iteratively in order of the indices. Let $K_i$ be the obtained graph after $i$ iterations. Since applications of Rule 1 do not increase degrees of vertices, each $K_i$ is degree-4-bounded. We now show by induction that the following invariant is true. For each $i \geq 0$, $K_i$ is a cactus and for each degree-4 vertex $v$ of $K_i$, there are arcs $(v, u), (v, u')$ with $u$ and $u'$ in different blocks. Intuitively, the invariant ensures that each $K_i$ is connected and that we can continue to apply Rule 1 until we have formed a Hamiltonian tour.

Initially, the invariant is true since $K_0 = K'$ is a cactus and we oriented each cycle of $K'$ in one direction. Now suppose that $1 \leq i \leq j$ and that the invariant is true for $K_{i-1}$. Let $v'$ be the vertex such that $K_i$ is obtained by applying Rule 1 to $v'$ in $K_{i-1}$. By the invariant, there are two blocks $C, C'$ of $K_{i-1}$ and vertices $w, w'$ such that $w \in V(C)$, $w' \in V(C')$, and $(v', w), (v', w') \in E(K_{i-1})$. Then $C, C'$ are edge disjoint cycles and thus removing both $(v', w)$ and $(v', w')$ leaves a connected graph. The new arc $\{w^-, w'^-\}$ closes a cycle with vertex set $V(C) \cup V(C')$, which implies that $K_i$ is 2-edge connected. We conclude that $K_i$ is a cactus, since an edge contained in two different cycles of $K_i$ would also be contained in two different cycles of $K_{i-1}$.

Each degree-4 vertex $v \neq v'$ of $K_{i-1}$ is also a cut vertex in $K_i$ since removing $v$ would leave two components only one of which containing the arcs changed by applying Rule 1. In particular, the property that there are arcs $(v, u), (v, u')$ with $u$ and $u'$ in different blocks of $K_i$ did not change. Thus the invariant also holds for $K_i$.

We conclude that $K''$ is a cactus with only vertices of degree 2 and therefore a Hamiltonian tour. □

To see that $\text{A}\textsc{lg}_\beta$ runs in polynomial time, observe that each separate step can be done in polynomial time. The iterated application of Rule 1 in the beginning and the end is done at most once for each vertex. Similarly, the while loop and the for loop are run at most linearly often.

We finish the proof by analyzing the approximation ratio of $\textsc{Alg}_\beta$. We aim to partition the edges of $H$ into paths such that for each edge $e$ of $K''$ there is a path in $H$ between the ends of $e$. To avoid ambiguity, in the following we fix $K'$ to be the version at the end of $\textsc{Alg}_\beta$.

We define a family $\mathcal{P}'$ of classes $P'_e \subseteq E(H)$, for each $e \in E(K')$. If $e \in E(H)$ we set $P'_e = \{e\}$. Otherwise, if $e \notin E(H)$, $e$ must have been obtained by applying Rule 1, replacing two edges $e', e''$ and we set $P'_e = \{e', e''\}$.

We claim that $\mathcal{P}'$ is a partition of $E(H)$. Let us fix an edge $e' \in E(H)$. Then either $e' \in E(K')$ and thus $P'_{e'} = \{e'\}$, or there is an $e''$ such that $e', e''$ were replaced by $e$. But then $e \in E(K')$ since during the while loop, eventually $e = \{u^-, v^-\}$ for two vertices $u, v$ and thus Rule 1 cannot replace $e$. Therefore $P'_e = \{e', e''\}$. Since during the while loop $e''$ also cannot have two heads, $e'' \in E(H)$. Furthermore, $e'$ uniquely determines its class in $\mathcal{P}'$ and therefore it cannot be in two different classes.

Analogously, we define a family $\mathcal{P}''$ of classes $P''_e \subseteq E(K')$, for each $e \in E(K'')$. If $e \in K'$, we set $P''_e = \{e\}$. Otherwise, if $e \notin E(K')$, $e$ must have been obtained by applying Rule 1, replacing two edges $e', e''$ and we set $P''_e = \{e', e''\}$. With the same arguments as for $\mathcal{P}'$, $\mathcal{P}''$ is a partition of $E(K')$.

We now define a partition $\mathcal{P}$ of $E(H)$ by combining $\mathcal{P}'$ and $\mathcal{P}''$. For each edge $e \in E(K'')$, we define the class $P_e$ of $\mathcal{P}$ as $\bigcup_{e' \in P''_e} P'_{e'}$.

Let $\tilde{E}$ be the set of all edges that in some iteration of the for loop of $\textsc{Alg}_\beta$ were denoted by $e$. Analogously, let $\tilde{E}'$ be the set of all edges that in some iteration of the for loop of $\textsc{Alg}_\beta$ were denoted by $e'$.

**Claim 3.6.** *For each $f \in E(K'') \setminus E(K')$, there are vertices $s, v, t$ such that $P''_f = \{\{s, v\}, \{v, t\}\}$ with $\{s, v\} \in E(H)$, $\{v, t\} \in \tilde{E}$, and $f = \{s, t\}$.*

*Proof.* Since $f \notin K'$, $f$ was obtained by applying Rule 1 to a vertex $v$. Therefore $P''_f = \{\hat{f}, e\} \subseteq E(K')$ where $\hat{f}$ and $e$ have $v$ as common vertex (see Fig. 2). Due to the orientation of $K'$ obtained during the for loop and in the subsequent step, there are two blocks $C' \neq C''$ of $K'$ such that for each of them, there is exactly one arc from $v$ into the block. These two arcs are $\hat{f}, e$, and we may assume that $\hat{f} \in E(C')$ and $e \in E(C'')$.

For one of the two blocks, $v$ is its exit point and by renaming we assume that $C''$ is that block. Therefore in $K$ (after the while loop), the two arcs $\hat{f}, \hat{g} \in E(C')$ incident to $v$ have tails at $v$. In particular, $\hat{f}$ was not obtained from an application of Rule 1 since the application only introduces arcs with two heads and thus $\hat{f} \in E(H)$. Due to the orientation in the for loop of $\textsc{Alg}_\beta$, we conclude that $e \in \tilde{E}$. The claim follows by fixing $s, t$ such that $\hat{f} = \{s, v\}$ and $e = \{v, t\}$. □

We now show that $\mathcal{P}$ partitions $E(H)$ into short paths.

**Claim 3.7.** *For each $f = \{s, t\} \in E(K'')$, the edges in $P_f$ form a path between $s$ and $t$ of length at most three in $H$.*

*Proof.* Let us first assume that $f \in E(K')$ Then either $f \in E(H)$ or $P_f = \{f', f''\}$, where $f' = \{s, v\}$ and $f'' = \{v, t\}$ for some vertex $v$. Since $f', f'' \in E(H)$, the claim follows.

Otherwise, if $f \notin E(K')$, there are edges $\hat{f}, e \in P''_f$ where $\hat{f} = \{s, v\}$ and $e = \{v, t\}$ for some vertex $v$. By Claim 3.6 we can rename the vertices and edges such that $\hat{f} \in E(H)$ and $e \in \tilde{E}$ (see Fig. 2). Then either $e \in E(H)$ or there is a vertex $u$ such that $P'_e = \{\{v, u\}, \{u, t\}\}$ with $\{v, u\}, \{u, t\} \in E(H)$. In both cases the claim follows. □

We now analyze the edge weights relative to $c(H)$. For edges not in $H$ we have to consider a factor that is bounded due to the relaxed triangle inequality. Let us fix an edge $f \in E(K'')$. We distinguish the sizes of $P_f$.

If $|P_f| = 1$, $f \in E(H)$ and thus there is no factor to be considered.

If $|P_f| = 2$, $P_f = \{f', f''\}$ for two adjacent edges $f', f'' \in E(H)$ and by the relaxed triangle inequality, $c(f) \leq \beta(c(f') + c(f''))$.

If $|P_f| = 3$, by Claim 3.6 we have that $c(f) \leq \beta \cdot (c(\hat{f}) + c(e))$ for an $\hat{f} \in E(H)$ and an $e \in \tilde{E}$ with $P''_f = \{\hat{f}, e\}$ (see Fig. 2). We define $\Pi \subseteq \tilde{E} \times \tilde{E}'$ such that $(\hat{e}, \hat{e}') \in \Pi$ if $\hat{e}$ and $\hat{e}'$ were compared in the for loop of $\text{ALG}_\beta$. Observe that since in the for loop each cycle is oriented only once, $\Pi$ assigns each edge from $E$ to exactly one edge of $E'$ and vice versa. Let $e' \in \tilde{E}'$ be the edge such that $(e, e') \in \Pi$. We have that $c(e) \leq \beta \cdot \min\{c(P'_e), c(P'_{e'})\}$, since otherwise $c(e') < c(e)$. Therefore, we obtain $c(f) \leq \beta c(\hat{f}) + \beta^2 \min\{c(P'_e), c(P'_{e'})\}$.

By Claim 3.7, we do not have to consider $|P_f| > 3$ and therefore there are no further cases.

In order to compute the overall cost of the solution, we define

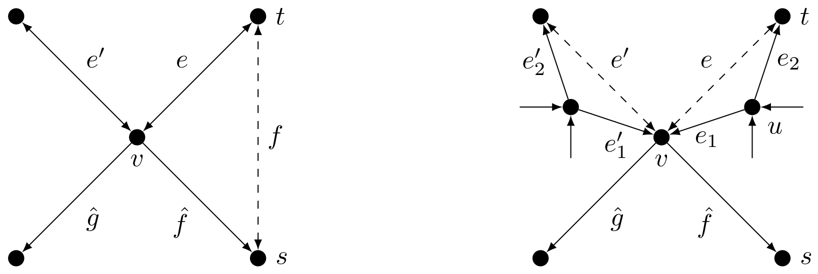
$$\begin{aligned} M := \{\hat{e} \in \tilde{E} : (\hat{e}, \hat{e}') \in \Pi \text{ and } c(P'_{\hat{e}}) \leq c(P'_{\hat{e}'})\} \\ \cup \{\hat{e} \in \tilde{E}' : (\hat{e}', \hat{e}) \in \Pi \text{ and } c(P'_{\hat{e}}) < c(P'_{\hat{e}'})\}, \end{aligned}$$

and $M' := (\tilde{E} \cup \tilde{E}') \setminus M$. Since we aim for edges in $E(H)$, $M_H := \bigcup_{g \in M} P'_g$, $M'_H := \bigcup_{g' \in M'} P'_{g'}$ and the remaining edges are $R := E(H) \setminus (M_H \cup M'_H)$.

Since $\mathcal{P}$ is a partition of $E(H)$, the overall cost $c(K'')$ is bounded from above by $\beta c(R) + \beta c(M'_H) + \beta^2 c(M_H)$. Since $c(M_H) \leq c(M'_H)$, we obtain that $c(K'') \leq \beta c(H)/2 + \beta^2 c(H)/2 \leq (3\beta/4 + 3\beta^2/4) \cdot \text{opt}$, where opt is the weight of an optimal $\beta$-TSP solution in $G$ and the last inequality follows from Theorem 3.1.

# 4 Conclusion

We have seen that in a complete edge-weighted graph we can obtain a degree-4-bounded Eulerian graph that has at most 1.5 times the weight of an optimal TSP solution. This result might allow for improvement. However, by setting $\beta = 1$ an improvement would directly imply an improvement over the approximation



(a) (b)

Figure 2: (a) A degree 4 vertex $v$ in $K$ after the while loop (solid arcs), where $f \in E(K'')$ is the arc that replaces $e$ and $\hat{f}$ when applying Rule 1 to $v$. (b) The same vertex $v$ and arcs from $H$ (solid) with the orientation before applying Rule 1 the first time. We have $e \in \tilde{E}$ and $e' \in \tilde{E}'$, i. e., $c(e) \leq c(e')$ and $c(f) \leq \beta(c(\hat{f}) + c(e))$. Furthermore, $P'_e = \{e_1, e_2\}$, $P'_{e'} = \{e'_1, e'_2\}$, $P''_f = \{\hat{f}, e\}$, and $P_f = \{\hat{f}, e_1, e_2\}$. Therefore $c(e) \leq \beta c(P'_e)$ and $c(e) \leq c(e') \leq \beta c(P'_{e'})$. Possibly $c(P'_{e'}) < c(P'_e)$, which implies $e' \in M$, $e \in M'$.

ratio of 1.5 of the metric traveling salesman problem and thus it would require valuable new insights.

The main result shows an approximability of the TSP with relaxed triangle inequality that is better than all previous results as long as the value of $\beta$ is smaller than 13/3. The most likely improvement of our result is for the term $3\beta^2/4$, as it is not linear in $\beta$. We conjecture the following.

**Conjecture 4.1.** *There is a polynomial time* $(1.5\beta)$*-approximation algorithm for* $\beta$*-TSP.*

Such a result seems likely as the $4\beta$-approximation algorithm does not include a factor $\beta^2$ and the factor $\beta^2$ in our result is only required in specific cases. Thus both proving and disproving the conjecture would lead to interesting insights.

# Acknowledgment

I would like to thank András Sebő for valuable pointers to the literature.

# References

[1] N. Christofides, Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem, Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, 1976.

[2] S. O. Gharan, A. Saberi, M. Singh, A randomized rounding approach to the traveling salesman problem, in: R. Ostrovsky (Ed.), Proc. of the 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011), IEEE, 2011, pp. 550–559.

[3] T. Mömke, O. Svensson, Approximating graphic TSP by matchings, in: R. Ostrovsky (Ed.), Proc. of the 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011), IEEE, 2011, pp. 560–569.

[4] M. Mucha, 13/9-approximation for graphic TSP, Theory of Computing Systems 55 (2012) 640–657.

[5] A. Sebő, J. Vygen, Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs, Combinatorica 34 (2014) 597–629.

[6] H.-J. Bandelt, Y. Crama, F. C. R. Spieksma, Approximation algorithms for multi-dimensional assignment problems with decomposable costs, Discrete Applied Mathematics 49 (1994) 25–50.

[7] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, W. Unger, Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem, Theoretical Computer Science 285 (2002) 3–24.

[8] T. Andreae, H.-J. Bandelt, Performance guarantees for approximation algorithms depending on parameterized triangle inequalities, SIAM Journal on Discrete Mathematics 8 (1995) 1–16.

[9] T. Andreae, On the traveling salesman problem restricted to inputs satisfying a relaxed triangle inequality, Networks 38 (2001) 59–67.

[10] M. Bender, C. Chekuri, Performance guarantees for TSP with a parametrized triangle inequality, Information Processing Letters 73 (2000) 17–21.

[11] T. Király, L. C. Lau, M. Singh, Degree bounded matroids and submodular flows, Combinatorica 32 (2012) 703–720.

[12] M. Singh, L. C. Lau, Approximating minimum bounded degree spanning trees to within one of optimal, in: D. S. Johnson, U. Feige (Eds.), Proc. of the 39th Annual ACM Symposium on Theory of Computing (STOC 2007), ACM Press, 2007, pp. 661–670.

[13] A. Schrijver, Combinatorial Optimization, Springer, 2003.

[14] J. Edmonds, E. L. Johnson, Matching, Euler tours and the Chinese postman, Mathematical Programming 5 (1973) 88–124.

[15] M. Held, R. M. Karp, The traveling-salesman problem and minimum spanning trees, Operations Research 18 (1970) 1138–1162.